

LAPORAN JOBSHEET 6

Dasar Pemrograman

Perulangan 1

Dibina oleh Ibu Vivi Nur Wijyaningrum, S.Kom., M.Kom.



Disusun oleh :

Fitria Ramadhani Prihandiva

D-IV Sistem Informasi Bisnis 1C / 14

2241760055

POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

TAHUN 2023

Nama : Fitria Ramadhani Prihandiva
Kelas : SIB 1C
NIM : 2241760055
Mata Kuliah : Praktikum Dasar Pemrograman

JOBSHEET 6

Percobaan 1 : Menghitung Bilangan Kelipatan Menggunakan FOR

```
1 import java.util.Scanner;
2 public class ForKelipatan1C14 {
3     public static void main(String[] args){
4         Scanner scan = new Scanner (System.in);
5         int kelipatan;
6         int jumlah=0;
7         int counter=0;
8
9         System.out.print("Masukkan bilangan kelipatan (1-9):");
10        kelipatan = scan.nextInt();
11
12        for (int i=1; i <= 50; i++){
13            if (i % kelipatan == 0) {
14                jumlah += i;
15                counter++;
16            }
17        }
18        System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);
19        System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %d\n", kelipatan, jumlah);
20    }
21 }
```

```
PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> java ForKelipatan1C14.java
Masukkan bilangan kelipatan (1-9):5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275
PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> █
```

Pertanyaan

1. Terdapat tiga komponen perulangan pada sintaks FOR. Berdasarkan Percobaan 1 tersebut, sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program yang telah dibuat!

for (int i = 1; i <= 50 ; i++)

int i = 1

Merupakan inisialisasi, sebelum memulai perulangan kita harus menentukan dari mana perulangan tersebut akan dimulai. Pada sintaks diatas perulangan diberi nama **i** dengan tipe data integer dan diberi inisialisasi **1**. Yang artinya pengulangan dimulai dari angka 1

i <= 50

Merupakan syarat, perulangan For digunakan untuk Definite Loop yaitu perulangan yang telah diketahui batas akhirnya. Sehingga fungsi syarat tersebut adalah untuk membatasi sampai mana perulangan tersebut akan di jalankan. Pada sintaks diatas perulangan diakhiri dengan syarat **i** merupakan angka kurang dari sama dengan 50 (1-50). Selain angka tersebut maka sintaks tidak akan dieksekusi/dijalankan

i++

Merupakan update, dimana perintah ini digunakan pada fungsi yang dinilai TRUE maka angka angka yang dimasukkan (inputan) akan diupdate terus menerus (Looping) hingga batas syarat yang telah ditentukan. Sedangkan apabila telah mencapai batas maka akan dinilai FALSE karena tidak memenuhi syarat yang sudah di masukkan sebelumnya, sehingga program akan

terhenti. Pada sintaks diatas perulangan yang dinilai benar akan di update dengan menambah +1 untuk tiap angka yang lolos(TRUE) pengecekan syarat.

2. Jelaskan alur kerja dari potongan kode program berikut!

```
for (int i = 1; i <= 50; i++) {  
    if (i % kelipatan == 0) {  
        total += i;  
        counter++;  
    }  
}
```

```
    If ( i % kelipatan ==0) {  
        total +=1;  
        counter++;  
    }
```

Sebelum memulai sintaks tersebut kita akan memasukan bilangan pada variabel **kelipatan** kemudian fungsi tersebut akan dieksekusi.

Jika **i** dalam sintaks diatas (Semua angka antara 1-50) di modulo oleh **kelipatan** yang kita masukkan sebelumnya akan menghasilkan 0 maka fungsi total dan counter akan dieksekusi.

Total = adalah jumlah total dari kelipatan yang ditemukan. Misalnya kita memasukkan 5 sebagai kelipatan maka akan ditemukan 5,10,15,20,25,30,35,40,45,50. Nah fungsi total akan menjumlahkan bilangan tersebut sehingga hasilnya 275.

Counter = adalah jumlah dari berapa kelipatan 5 dari 1 hingga 50. Misalnya kita memasukkan 5 sebagai kelipatan maka akan ditemukan 10 bilangan yang merupakan kelipatan dari 5 pada range 1-50.

3. Modifikasi kode program yang telah dibuat dengan menambahkan variabel baru untuk menghitung rata-rata dari seluruh bilangan kelipatan yang ditentukan!

```
1  import java.util.Scanner;  
2  public class ForKelipatan1C14 {  
3      public static void main(String[] args){  
4          Scanner scan = new Scanner (System.in);  
5          int kelipatan;  
6          double jumlah=0;  
7          int counter=0;  
8          double rata;  
9  
10         System.out.print("Masukkan bilangan kelipatan (1-9):");  
11         kelipatan = scan.nextInt();  
12  
13         for (int i=1; i <= 50; i++){  
14             if (i % kelipatan == 0) {  
15                 jumlah += i;  
16                 counter++;  
17             }  
18         }  
19         rata = jumlah/counter;  
20         System.out.printf("Banyaknya bilangan %d dari 1 sampai 50 adalah %d\n", kelipatan, counter);  
21         System.out.printf("Total bilangan kelipatan %d dari 1 sampai 50 adalah %.1f\n", kelipatan, jumlah);  
22         System.out.printf("Rata rata dari seluruh bilangan %d kelipatan yang ditemukan adalah %.1f\n", kelipatan, rata);  
23     }  
24 }
```

Saya memodifikasi dengan menambahkan variable bertipe **double** dengan nama **rata**. Kemudian merubah tipe data **jumlah** pada deklarasi menjadi **double**

Selanjutnya saya menambahkan perhitungan diluar sintaks **FOR** yaitu **rata= jumlah/counter**.

Selanjutnya saya merubah **%d\n** menjadi **%.1f\n** agar bisa memunculkan angka desimal dengan satu angka saja dibelakang koma.

Menambahkan satu lagi sintaks **printf**. Untuk menampilkan rata rata yang telah terhitung.

Berikut adalah hasil output nya:

```
PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> java ForKelipatan1C14.java
Masukkan bilangan kelipatan (1-9):5
Banyaknya bilangan 5 dari 1 sampai 50 adalah 10
Total bilangan kelipatan 5 dari 1 sampai 50 adalah 275.0
Rata rata dari seluruh bilangan 5 kelipatan yang ditemukan adalah 27.5
```

Percobaan 2 : Menghitung Gaji Lembur Karyawan Menggunakan WHILE dan Continue

```
1  import java.util.Scanner;
2  public class WhileGaji1C14 {
3      public static void main(String[] args){
4          Scanner scan = new Scanner (System.in);
5          int karyawan, jam, gaji;
6          int jumlah =0;
7          String jabatan;
8
9          System.out.print("Masukkan jumlah karyawan: ");
10         karyawan = scan.nextInt();
11
12         int i=0;
13         while (i<karyawan) {
14             System.out.println("Pilihan jabatan - Direktur, Manager, Staf");
15             System.out.print("Masukkan jabatan karyawan ke-" + (i+1) + ":");
16             jabatan = scan.next();
17             System.out.print("Masukkan jumlah jam lembur: ");
18             jam = scan.nextInt();
19             i++;
20             if (jabatan.equalsIgnoreCase("direktur")){
21                 continue;
22             }else if (jabatan.equalsIgnoreCase("manager")){
23                 gaji = jam * 100000;
24             }else {
25                 gaji = jam * 75000;
26             }
27             jumlah += gaji;
28         }
29         System.out.print("Pengeluaran perusahaan = " + jumlah);
30     }
31 }
```

```
PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> java WhileGaji1C14.java
Masukkan jumlah karyawan: 3
Pilihan jabatan - Direktur, Manager, Staf
Masukkan jabatan karyawan ke-1:manager
Masukkan jumlah jam lembur: 1
Pilihan jabatan - Direktur, Manager, Staf
Masukkan jabatan karyawan ke-2:direktur
Masukkan jumlah jam lembur: 20
Pilihan jabatan - Direktur, Manager, Staf
Masukkan jabatan karyawan ke-3:staf
Masukkan jumlah jam lembur: 10
Pengeluaran perusahaan = 850000
```

1. Tunjukkan bagian kode program yang digunakan sebagai syarat untuk menghentikan perulangan WHILE! Berapa kali perulangan dilakukan?

Syarat ada pada sintaks **while (i<karyawan)** perulangan terjadi sebanyak kita memasukkan jumlah pada variabel **karyawan**. Dengan Syarat jumlah **karyawan** harus lebih besar dari **i**. Sedangkan **i** telah diinisialisasi dengan jumlah 0. Sehingga dapat disimpulkan bahwa input jumlah **karyawan** hanya bisa dijalankan apabila berjumlah positif (diatas 0)

```
int i=0;
while (i<karyawan) {
```

2. Pada potongan kode berikut,

```
if (jabatan.equalsIgnoreCase("direktur")) {
    continue;
```

Apa yang sebenarnya terjadi jika variabel **jabatan** berisi nilai "DIREKTUR"? Apa peran CONTINUE yang dituliskan di dalam sintaks perulangan?

Pada saat jabatan **direktur** dimasukkan maka tidak ada perhitungan **gaji** yang di eksekusi. Penggunaan fungsi **Continue** adalah bertujuan untuk melanjutkan pada sintaks selanjutnya. Sehingga apabila **direktur** dimasukkan maka program akan langsung berlanjut mengeksekusi sintaks yang selanjutnya yaitu jabatan **manager**.

3. Mengapa komponen update **i++** diletakkan di posisi tengah, tidak di bagian akhir statement? Pindahkan **i++** di bagian akhir, lalu jalankan kembali program dengan memasukkan "direktur" sebagai jabatan karyawan pertama. Apa yang terjadi? Jelaskan!

```
int i=0;
while (i<karyawan) {
    System.out.println("Pilihan jabatan - Direktur, Mana
    System.out.print("Masukkan jabatan karyawan ke-" + (
    jabatan = scan.next());
    System.out.print("Masukkan jumlah jam lembur: ");
    jam = scan.nextInt();

    if (jabatan.equalsIgnoreCase("direktur")){
        continue;
    }else if (jabatan.equalsIgnoreCase("manager")){
        gaji = jam * 100000;
    }else {
        gaji = jam * 75000;
    } i++;
    jumlah += gaji;
}
System.out.print("Pengeluaran perusahaan = " + jumlah);
```

Saya mencoba memindahkan **i++** pada akhir sintaks **while**. Outputnya sebagai berikut

```

PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> java WhileGaji1C14.java
Masukkan jumlah karyawan: 3
Pilihan jabatan - Direktur, Manager, Staf
Masukkan jabatan karyawan ke-1: direktur
Masukkan jumlah jam lembur: 5
Pilihan jabatan - Direktur, Manager, Staf
Masukkan jabatan karyawan ke-1: manager
Masukkan jumlah jam lembur: 5
Pilihan jabatan - Direktur, Manager, Staf
Masukkan jabatan karyawan ke-2: staf
Masukkan jumlah jam lembur: 3
Pilihan jabatan - Direktur, Manager, Staf
Masukkan jabatan karyawan ke-3: staf
Masukkan jumlah jam lembur: 5
Pengeluaran perusahaan = 1100000

```

Perulangan terjadi sebanyak 4x dan ketika **jabatan direktur** dimasukkan masih tertulis (karyawan ke 1) dan setelah selesai, perulangan masih tertulis (karyawan ke 1). Baru ketika memasukkan **manager** dan **staf** maka penulisan karyawan ke- terupdate menjadi karyawan ke 2 dan seterusnya. Hingga terjadi 3x pengulangan.

Alasan diletakkan di tengah adalah agar pengulangan direktur tetap terhitung meski bertemu continue jumlah **i** tetap terupdate.

Percobaan 3 : Menghitung Jatah Cuti Menggunakan DO-WHILE dan BREAK

```

1  import java.util.Scanner;
2  public class DoWhileCuti1C14 {
3      public static void main(String[] args){
4          Scanner scan = new Scanner (System.in);
5          int cuti = 5;
6          char konfirmasi;
7
8          do {
9              System.out.print("Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?" );
10             konfirmasi = scan.next().charAt(0);
11             if (konfirmasi == 'y' || konfirmasi == 'Y'){
12                 cuti--;
13                 System.out.printf("Sisa %d hari\n", cuti);
14                 if (cuti == 2) {
15                     System.out.println("STOP! Jatah cuti tinggal 2 hari");
16                     break;
17                 }
18             }
19         }while (cuti>0);
20     }
21 }

```

```

PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> java DoWhileCuti1C14.java
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?y
Sisa 4 hari
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?Y
Sisa 3 hari
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?Y
Sisa 2 hari
STOP! Jatah cuti tinggal 2 hari
PS D:\College\SEMESTER 2\Daspro\Jobsheet 6>

```

```

PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> java DoWhileCuti1C14.java
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?t
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?y
Sisa 4 hari
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?y
Sisa 3 hari
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?y
Sisa 2 hari
STOP! Jatah cuti tinggal 2 hari
PS D:\College\SEMESTER 2\Daspro\Jobsheet 6>

```

Pertanyaan

1. Apa kegunaan sintaks BREAK di dalam sintaks perulangan?

Break digunakan untuk menghentikan program dalam artian untuk keluar dari proses looping.

2. Jalankan kembali program tersebut. Pada saat input konfirmasi, ketikkan “Tidak”, apa yang terjadi? Mengapa demikian?

```

ge\SEMESTER 2\Daspro\Jobsheet 6>
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?tidak
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?tidak
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?tidak
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?tidak
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?tidak
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?tidak
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?

```

Perulangan terjadi tanpa batas dikarenakan kita tidak memasukkan fungsi **else** setelah **if** konfirmasi y. Sehingga apapun yang dimasukkan selain huruf (y/Y) akan terus melakukan perulangan.

3. Modifikasi kode program sehingga saat pengguna mengetikkan “Tidak” sebagai input konfirmasi, maka program akan berhenti!

```

1  import java.util.Scanner;
2  public class DoWhileCuti1C14 {
3      public static void main(String[] args){
4          Scanner scan = new Scanner (System.in);
5          int cuti = 5;
6          char konfirmasi;
7
8          do {
9              System.out.print("Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?" );
10             konfirmasi = scan.next().charAt(0);
11             if (konfirmasi == 'y' || konfirmasi == 'Y'){
12                 cuti--;
13                 System.out.printf("Sisa %d hari\n", cuti);
14                 if (cuti == 2) {
15                     System.out.println("STOP! Jatah cuti tinggal 2 hari");
16                     break;
17                 }
18             }else {
19                 break;
20             }
21         }while (cuti>0);
22     }
23 }
24

```

Saya menambahkan fungsi **else** lalu di **break** agar program terhenti. Output sebagai berikut:

```

PS D:\College\SEMESTER 2\Daspro\Jobsheet 6> java DoWhileCuti1C14.java
Apakah Anda menggunakan jatah cuti 1 hari (y/t) ?Tidak

```